WebGuardian: Raspberry Pi and Arduino Powered Quad-Bot for Web Surveillance

Thesis Submitted to Central University of Karnataka, Kalaburagi
in partial fulfillment of the award of the degree of

Bachelor of Technology

In

Electronics and Communication Engineering

by

Pilli Venkata Praneeth

(Reg. No: 2019BEC22)

Under the Supervision of

Dr. Layak Ali

Department of Electronics and Communication Engineering,
School of Engineering,
Central University of Karnataka
Kadaganchi, Kalaburagi, India 585 367.

May 2023

CENTRAL UNIVERSITY OF KARNATAKA

(Established by on Act of the Parliament in 2009)
Kadaganchi, Kalaburagi - 585 367

Electronics and Communication Engineering
School of Engineering
Website: www.cuk.ac.in

CERTIFICATE OF THE SUPERVISOR/GUIDE

This is to certify that the thesis entitled **"WebGuardian: Raspberry Pi and Arduino Powered Quad-Bot for Web Surveillance"** submitted to Central University of Karnataka, Kalaburagi, India by **Mr. Pilli Venkata Praneeth,** Registration No: **2019BEC22**, in fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Electronics and Communication Engineering**, was done by her, under my guidance and supervision. The data, facts and figures reported in it are genuine. It represents her original work and is worthy of consideration for the award of Bachelor of Technology in Electronics and Communication Engineering.

Further declare that the thesis has no plagiarism and not been submitted earlier in part or full to this or any other university /institute/college/research body for the award of any degree/diploma.

Signature of Supervisor/Guide

Place: Kadaganchi
Date:

Dr. Layak Ali
Assistant Professor
Electronics and Communication Engineering
School of Engineering
Central University of Karnataka
Kadaganchi, 585 367.

CENTRAL UNIVERSITY OF KARNATAKA

(Established by on Act of the Parliament in 2009)
Kadaganchi, Kalaburagi - 585 367

Electronics and Communication Engineering
School of Engineering
Website: www.cuk.ac.in

CERTIFICATE OF THE DEAN/ HEAD OF THE DEPARTMENT

This is to certify that the thesis entitled **"WebGuardian: Raspberry Pi and Arduino Powered Quad-Bot for Web Surveillance"** submitted to Central University of Karnataka, Kalaburagi, India by **Mr. Pilli Venkata Praneeth,** Registration No: **2019BEC22**, in fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Electronics and Communication Engineering**. This thesis has not been submitted to any university /institute/college/research body for the award of any degree diploma or certificate.

Signature of Dean/HOD

Place: Kadaganchi

Date:

Dr. Veeresh Gangappa Kasabegaudar
Dean, School of Engineering
Central University of Karnataka
Kadaganchi, 585 367.

DECLARATION OF THE STUDENT

I hereby declare that the work reported in the B . Tech. thesis entitled **"WebGuardian: Raspberry Pi and Arduino Powered Quad-Bot for Web Surveillance**" submitted to Central University of Karnataka, Kalaburagi, India is an authentic record of my work carried out under the supervision of **Dr. Layak Ali**. It is further certified that there is no plagiarism and this work has not been submitted for the award of any other degree/ diploma of this University or any other institution. I further attest that this work is original and that I am fully responsible for the content of my B. Tech thesis.

(Signature of the Student)

Pilli Venkata Praneeth

Place: Kadaganchi

Date:

ACKNOWLEDGEMENTS

I wish to express my profound gratitude to the Vice Chancellor **Prof.Battu Satyanarayana**, for his constant support and also for providing all the facilities in the university. I would like to express my sincere thanks to **Dr. Veeresh Gangappa Kasabegaudar**, Dean and Head of the department of Electronics and Communication Engineering for his continuous support and encouragement. I feel deeply indebted to my esteemed Project Coordinator and Guide, **Dr. Layak Ali** for their guidance, right from the conception and visualisation to the very presentation of the project. I am greatly indebted to our faculties, both teaching and supporting staff, Department of Electronics and Communication Engineering, who took great interest in my project work. They motivated and guided me throughout the accomplishment of this goal. I express my profound thanks for their meticulous guidance. Finally, I would like to express my heartfelt thanks to my beloved **parents** for their blessings, my **friends** for their help and wishes for the successful completion of this project work.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

TERM	ABBREVITION
LED	Light Emitting Diode
RX	Receiver
TX	Transmitter
IDE	Integrated Development Board
USB	Universal Serial Bus
CPU	Central Processing Unit
GND	Ground
VCC	Voltage Common Collector
DC	Direct Current
ICSP	In Circuit Serial Programming

Dedicated to

My Parents, Family Members, Professors and Friends who always stood by me and provided guidance in pursuing this work

CHAPTER-1
INTRODUCTION

1.1 General

The "Web-Guardian" is an innovative project that combines the power of Raspberry Pi and Arduino Uno to create a versatile quad-bot for web surveillance. This project aims to develop a web-based surveillance system that can remotely monitor and protect an area using a quad-bot equipped with a webcam or Raspberry Pi Camera Module. By leveraging the Raspberry Pi's web connectivity and image processing capabilities, along with the Arduino Uno's motor control functionality, the Web-Guardian quad-bot can provide real-time video streaming and surveillance features. The primary objective of this project is to design a robust surveillance system that can be controlled and monitored remotely through a web interface. The quad-bot's mobility allows it to navigate various terrains and capture video footage from different perspectives. By implementing image processing algorithms, such as motion detection or object recognition, the quad-bot can intelligently identify and track potential intrusions or suspicious activities.

The Raspberry Pi acts as the central processing unit for the project, handling web connectivity, video capture, and image processing tasks. The Arduino Uno, on the other hand, serves as the controller for the quad-bot's movement, providing precise motor control based on commands received from the Raspberry Pi. The integration of these two platforms enables seamless communication and coordination between the web interface, the surveillance algorithms, and the physical movements of the quad-bot. Through this project, users will have the ability to access the live video stream from the quad-bot remotely using any web-enabled device. They can control the quad-bot's movement, switch between different surveillance modes, and receive alerts or notifications when any suspicious activities are detected. This web-based surveillance system can find applications in various scenarios, including home security, outdoor monitoring, and remote area surveillance.

The Web-Guardian project combines the power of open-source technologies, the versatility of Raspberry Pi, and the control capabilities of Arduino Uno to create an effective and affordable solution for web-based surveillance. By building upon the foundation of these platforms, this project opens up opportunities for customization, further enhancements, and integration with other smart home or IoT systems. Overall, the Web-Guardian project offers an exciting exploration of the possibilities of combining Raspberry Pi and Arduino Uno to create a quad-bot for web surveillance, providing users with a reliable, accessible, and customizable solution for remote monitoring and protection.

1.2 Problem statement

Traditional surveillance systems often lack mobility, accessibility, and real-time monitoring capabilities. Fixed cameras or stationary systems may not provide comprehensive coverage or the ability to adapt to dynamic environments. Additionally, remotely accessing and controlling such systems can be challenging.

The problem this project aims to address is the need for an efficient and user-friendly web-based surveillance solution that offers mobility, real-time monitoring, and remote accessibility. By utilizing the power of Raspberry Pi and Arduino Uno, the project seeks to overcome the limitations of conventional surveillance systems.

The specific challenges to be addressed include:

1. Mobility: Developing a quad-bot that can traverse various terrains and provide surveillance from different perspectives, ensuring comprehensive coverage of the monitored area.
2. Real-time Monitoring: Implementing a system that can capture and stream live video footage, enabling real-time monitoring of the surveillance area.
3. Web Connectivity: Establishing a reliable web interface that allows users to remotely access the surveillance system, control the quad-bot's movements, and receive real-time alerts or notifications.
4. Image Processing: Implementing intelligent algorithms for image processing, such as motion detection or object recognition, to identify and track potential intrusions or suspicious activities.
5. Integration and Control: Ensuring seamless communication and coordination between the web interface, the image processing algorithms, and the quad-bot's movement controlled by Arduino Uno.

CHAPTER-II

LITRATURE SURVAY

Quadrupedal robots have garnered significant attention in the field of robotics due to their ability to navigate challenging terrains, mimic animal locomotion, and perform various tasks. This literature review provides an overview of key research contributions and notable inventors in the field of quadrupedal robotics.

- Raibert, Marc H. and Boston Dynamics:

  Marc H. Raibert, founder of Boston Dynamics, has made significant contributions to the development of dynamic quadrupedal robots. His work on the Legged Squad Support System (LS3) and BigDog showcased the capabilities of robust, agile quadrupedal locomotion.

- Kuffner, James and the ANYmal Robot:

  James Kuffner, along with the team at ETH Zurich and ANYbotics, has pioneered the development of the ANYmal robot. The ANYmal platform focuses on robust and versatile quadrupedal locomotion, allowing it to perform inspection and maintenance tasks in challenging environments.

- Kalouche, Ali and Bidaud, Philippe:

  Ali Kalouche and Philippe Bidaud have conducted extensive research in the design and control of quadruped robots. Their review paper titled "Design and control of quadruped robots: A review" provides insights into the various design considerations, control strategies, and applications of quadrupedal robots.

- Geyer, Hartmut and Seyfarth, Andre:

  Hartmut Geyer and Andre Seyfarth have contributed to the field of quadrupedal locomotion through their research on bio-inspired control and dynamics. Their work focuses on understanding and replicating the principles of animal locomotion to achieve efficient and stable quadrupedal motion.

- Coppelia Robotics and the V-REP Simulator:

Coppelia Robotics, a leading robotics software company, has developed the V-REP (Virtual Robot Experimentation Platform) simulator. V-REP allows researchers to simulate and experiment with various aspects of quadrupedal locomotion, control, and sensor integration.

- Stanford Doggo and the Stanford Robotics Club:

  The Stanford Robotics Club's Doggo project gained attention for its low-cost, open-source quadrupedal robot. Doggo offers an accessible platform for students and researchers to explore quadrupedal locomotion and develop novel control strategies.

CHAPTER- III

IMPLIMENTATION AND OVERVIEW OF SYSTEM

3.1 System overview

The Quad-Surveil project aims to develop a web-based surveillance system using Raspberry Pi and Arduino Uno to create a versatile quad-bot capable of remote monitoring and protection. The system combines hardware components, software modules, and a web interface to enable real-time video streaming, remote control, and intelligent surveillance features.

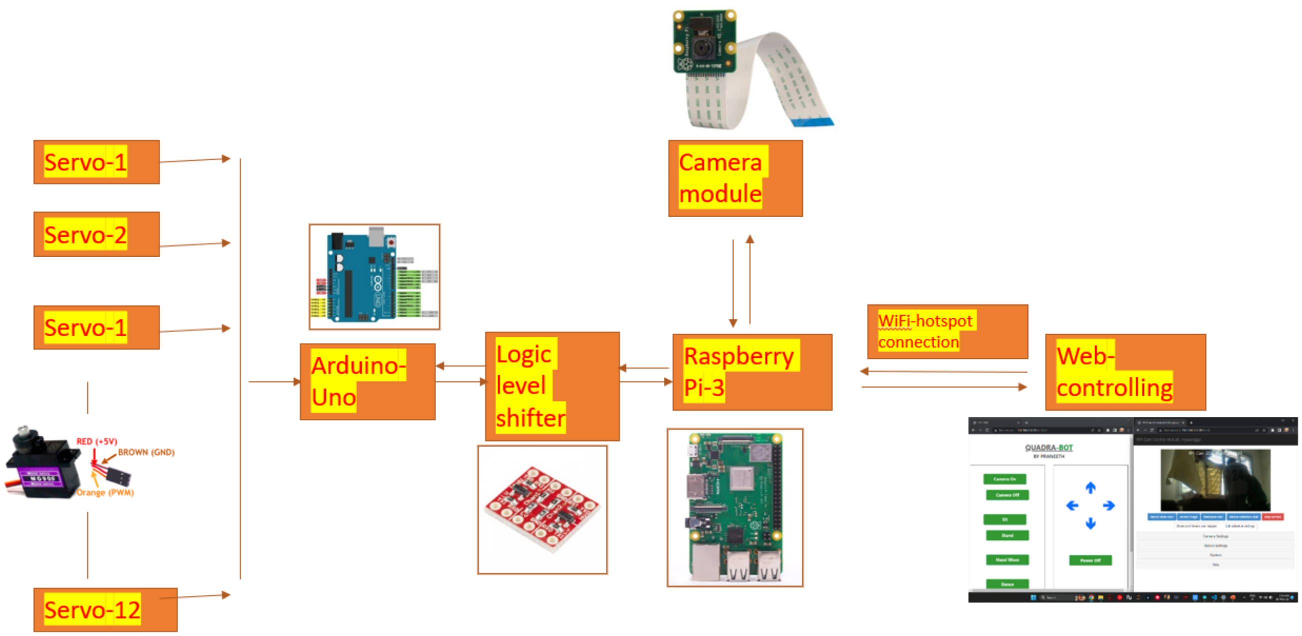


Fig 3.1 system block diagram

3.2 Hardware components

1. Quad-Bot Hardware:

   The quad-bot is built using a combination of Raspberry Pi and Arduino Uno. Raspberry Pi acts as the central processing unit and handles web connectivity, video capture, and image processing tasks. Arduino Uno is responsible for motor control and movement coordination.

i) Arduino Uno

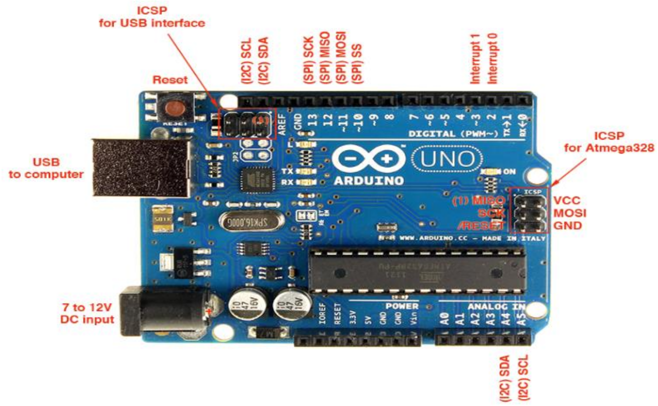


Fig 3.2 Arduino Uno

The Arduino Uno is a popular microcontroller board that offers a flexible and user-friendly platform for electronics prototyping and projects. Here are the specifications of the Arduino Uno:

Microcontroller: ATmega328P microcontroller

Operating Voltage: 5 volts

Digital I/O Pins: 14 digital I/O pins, including 6 pins capable of PWM output

Analog Input Pins: 6 analog input pins

DC Current per I/O Pin: 20 mA

DC Current for 3.3V Pin: 50 mA

Flash Memory: 32 KB (ATmega328P), of which 0.5 KB is used by the bootloader

SRAM: 2 KB (ATmega328P)

EEPROM: 1 KB (ATmega328P)

Clock Speed: 16 MHz

Communication: USB interface for programming and serial communication

TTL (Transistor-Transistor Logic) serial communication via the RX/TX pins

Power: Can be powered via USB connection or an external power supply (7-12V DC)

ii) Raspberry Pi 3 Model B+

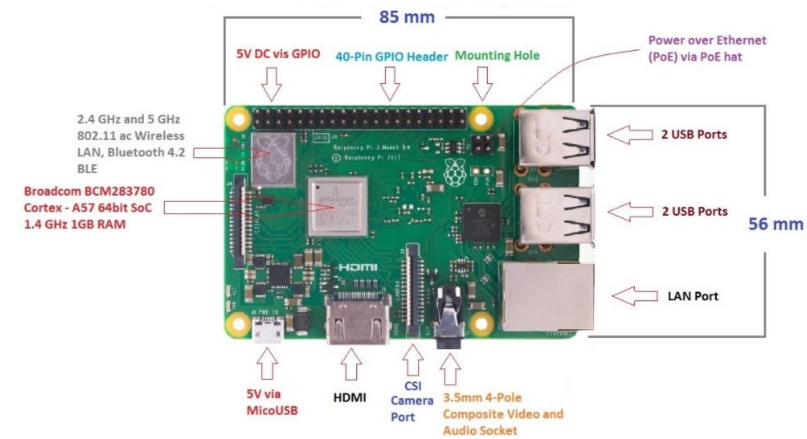


Fig 3.3 Raspberry Pi 3 Model B+

The Raspberry Pi 3 Model B+ is a single-board computer that offers a range of features and capabilities. Here are the specifications of the Raspberry Pi 3 Model B+:

Processor: Broadcom BCM2837B0 64-bit quad-core ARM Cortex-A53 processor

Clock speed: 1.4 GHz

Memory: 1GB LPDDR2 SDRAM

Storage: MicroSD card slot for the operating system and data storage

Connectivity: 2.4 GHz and 5 GHz IEEE 802.11.b/g/n/ac wireless LAN

Bluetooth 4.2/BLE

10/100/1000 Ethernet port (RJ45)

USB: 4 x USB 2.0 ports

Camera Interface: CSI camera port for connecting a Raspberry Pi camera module

GPIO: 40-pin GPIO header for connecting various external devices.

Power: Micro USB power connector (5V/2.5A power supply recommended)

Operating System: Compatible with various Linux distributions, including Raspbian (officially supported), Ubuntu, and others.

These specifications highlight the key features of the Raspberry Pi 3 Model B+, including its quad-core processor, wireless connectivity options, multiple USB ports, video and audio output capabilities, camera interface, GPIO header, and power requirements. It provides a versatile platform for a wide range of projects and applications.

2. Webcam or Raspberry Pi Camera Module:

   The quad-bot is equipped with a webcam or Raspberry Pi Camera Module to capture video footage of the surveillance area. The camera provides a live video feed that is processed and transmitted via the Raspberry Pi.

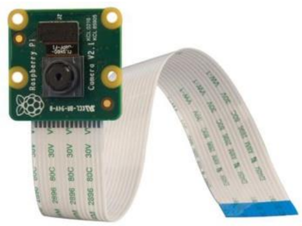


Fig 3.4 Raspberry Pi-camera

3. Web Interface:

   The web interface serves as the control center for the surveillance system. Users can remotely access the system using any web-enabled device such as a smartphone or computer. The web interface allows users to view the live video stream, control the quad-bot's movements, and switch between different surveillance modes.

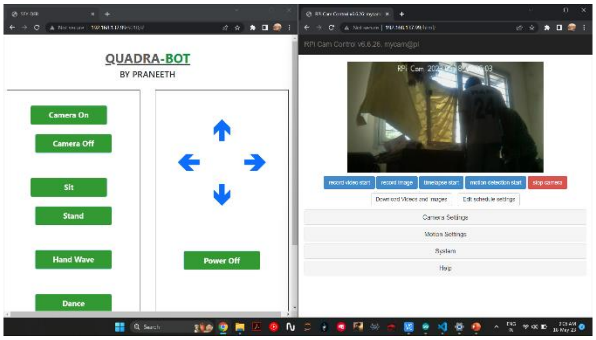


Fig 3.5 Web page interface

4. Real-Time Video Streaming:

The Raspberry Pi processes the video feed from the camera and enables real-time video streaming. The live video stream is transmitted over the internet to the web interface, allowing users to monitor the surveillance area remotely.

5. Image Processing:

   The system incorporates image processing algorithms, such as motion detection or object recognition, to analyze the video footage. These algorithms enable the quad-bot to identify potential intrusions or suspicious activities and trigger appropriate actions or alerts.

6. Motor Control and Movement:

   Arduino Uno is responsible for controlling the quad-bot's movements. It receives commands from the Raspberry Pi based on user inputs from the web interface. The Arduino controls the motors and actuators to navigate the quad-bot in the desired direction.

   a) MG90S servo motor

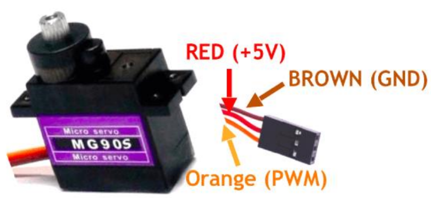


Fig 3.6 MG90S servo motor

The MG90S is a popular servo motor widely used in robotics, RC vehicles, and other applications. Here are the specifications of the MG90S servo motor:

Operating Voltage: 4.8V to 6.0V

Stall Torque:2.0 kg/cm (4.8V)

2.2 kg/cm (6.0V)

Operating Speed: 0.11 sec/60° (4.8V)

0.10 sec/60° (6.0V)

Control Angle: 180°

Gear Type: Metal gears for improved durability and precision

Motor Type: Coreless motor for smoother and more efficient operation

Dimensions: Size: 22.8mm x 12.2mm x 28.5mm

Weight: Approximately 13 grams

Operating Temperature: -30°C to +60°C

Connector Type: 3-pin standard servo connector (Signal, VCC, Ground)

The MG90S servo motor is known for its compact size, lightweight design, and reliable performance. It offers a decent amount of torque and speed for its size, making it suitable for various robotic and hobbyist applications. The metal gears enhance durability and ensure precise movement control. The servo motor operates within a voltage range of 4.8V to 6.0V, commonly provided by standard RC battery packs or power supplies.The MG90S servo motor's specifications make it compatible with a wide range of projects, including robotic arms, pan-tilt mechanisms, remote-controlled vehicles, and other motion control systems.

7. 2 Channel Logic Level Converter 3.3V to 5V

Fig 3.7 2 Channel Logic Level Converter

A logic level shifter is an electronic component used to interface devices that operate at different logic voltage levels. In the context of connecting an Arduino and a Raspberry Pi 3, a logic level shifter can be employed when the voltage levels of their respective input/output pins are different.

The Arduino Uno operates at 5V logic levels, meaning its digital HIGH state is represented by 5 volts, while the Raspberry Pi 3 operates at 3.3V logic levels, with its digital HIGH state represented by 3.3 volts. When connecting the two devices directly, without a level shifter, there is a risk of damaging the Raspberry Pi's GPIO pins, as they may not tolerate the higher voltage levels used by the Arduino.

To ensure proper communication and protect the Raspberry Pi's GPIO pins, a logic level shifter is used as an intermediary. The level shifter converts the 5V signals from the

Arduino to 3.3V signals that are safe for the Raspberry Pi. It typically consists of voltage translation circuitry, such as level-shifting ICs or bidirectional voltage translators, that can step down the voltage from 5V to 3.3V.

Overall, the Quad-Surveil project integrates Raspberry Pi, Arduino Uno, web connectivity, image processing, and motor control to create a web-based surveillance system. The system enables remote monitoring, real-time video streaming, intelligent surveillance, and user-friendly control through the web interface. It provides an accessible and customizable solution for enhanced security and monitoring in various settings.

CHAPTER- IV

3D DESIGNING AND PRINTING

4.1 3D designing

Starting by creating a new project in FreeCAD. Using FreeCAD's design tools to model the body of the quadruped robot.

Create and modify 3D shapes, extrude features, and add details to achieve the desired design.

Take into consideration the mechanical constraints, dimensions, and functional requirements of the quadruped robot.

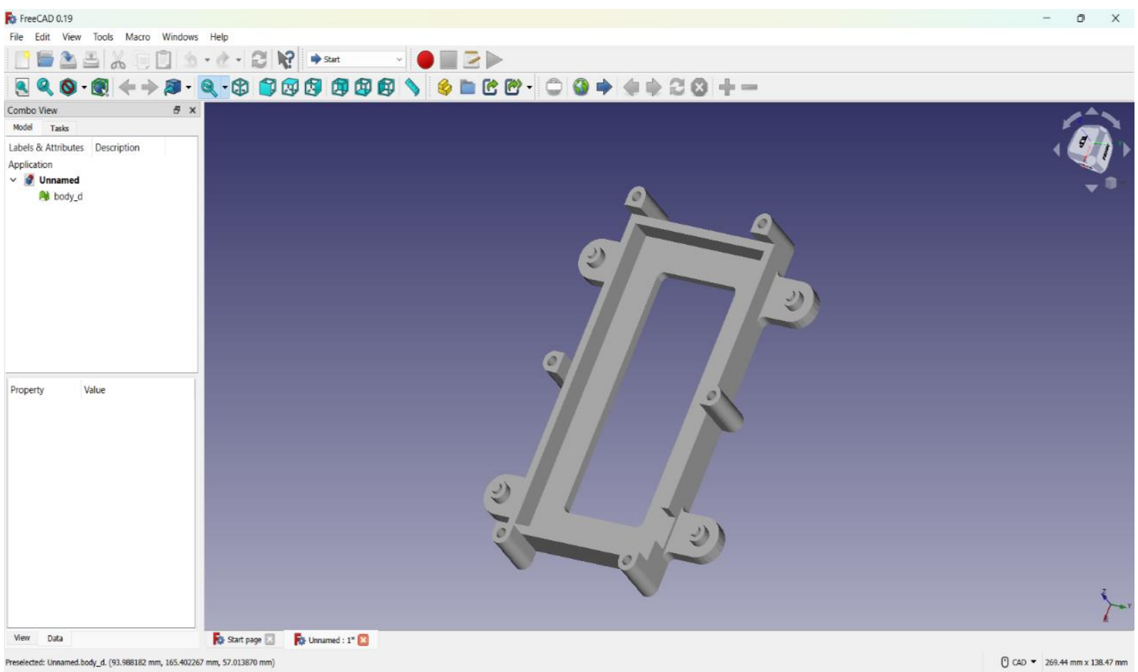


Fig 4.1 FreeCAD designing

4.2 Exporting the Design:

Once the design is complete, export it from FreeCAD in a format compatible with Cura Ultimaker. Common file formats include STL (Standard Tessellation Language) or OBJ (Wavefront Object).

Importing the Design into Cura Ultimaker:

Open Cura Ultimaker software.

Import the exported STL or OBJ file of the quadruped body design into Cura.

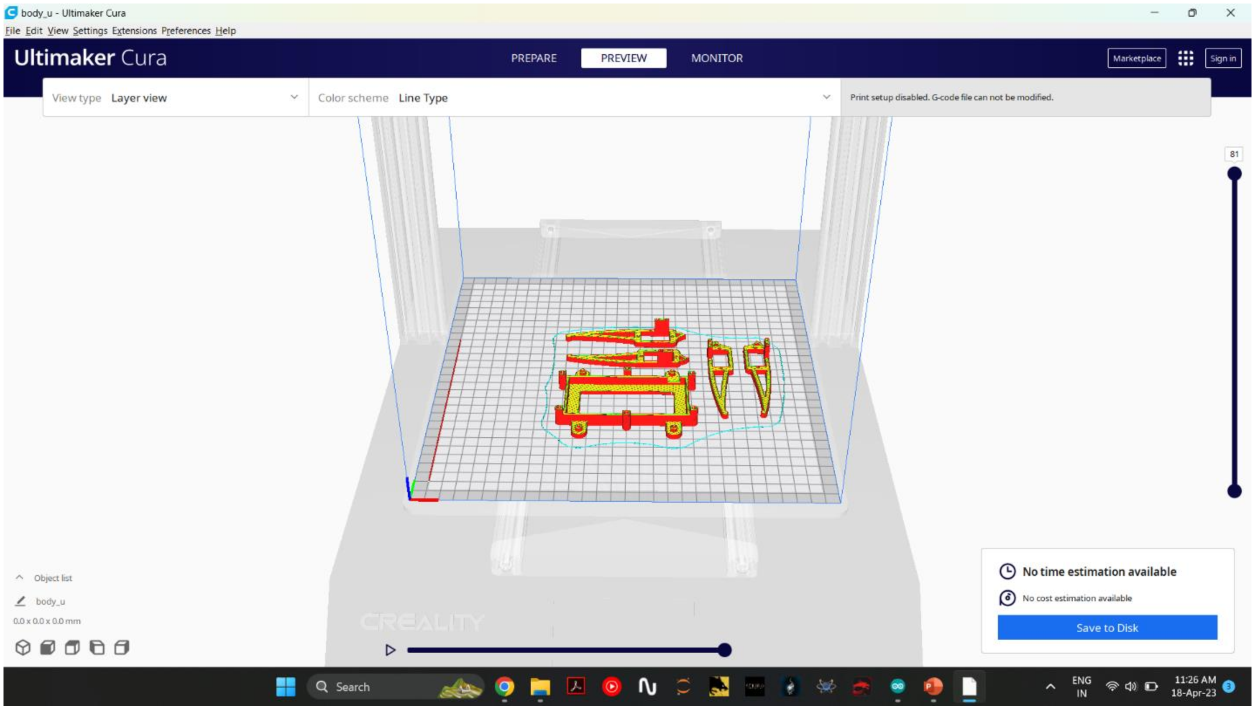


Fig 4.2 Importing stl files to Cura

Position and scale the model within the build area to fit the desired dimensions and orientation for printing.

Configuring Printer Settings:

Set up the printer profile in Cura Ultimaker for the Creality 10S Pro 3D printer.

Configure parameters such as print bed size, nozzle size, filament type, and printing temperature.

Ensure that the settings align with the specific characteristics and capabilities of the 3D printer being used.

Adjusting Printing Parameters:

When using Cura for 3D printing with PLA filament, consider the following settings. PLA typically prints well at temperatures between 185°C to 220°C. Start with around 200°C and adjust as needed. A heated build plate at 50°C to 60°C helps with adhesion, while non-heated plates can work at room temperature. A layer height of 0.1mm to 0.2mm provides good resolution. Moderate print speeds of around 50mm/s are suitable for PLA. Set the wall thickness to multiples of your nozzle diameter (e.g., 0.8mm for a 0.4mm nozzle). An infill density of 20% to 30% balances strength and material usage. Enable support structures as needed for overhangs. Active cooling with a fan enhances print quality and reduces warping. Retraction settings of around 6mm distance and 25mm/s to 40mm/s speed help prevent stringing. Remember that these are general guidelines, and it's important to experiment and adjust settings based on your specific printer, filament, and desired print outcome.

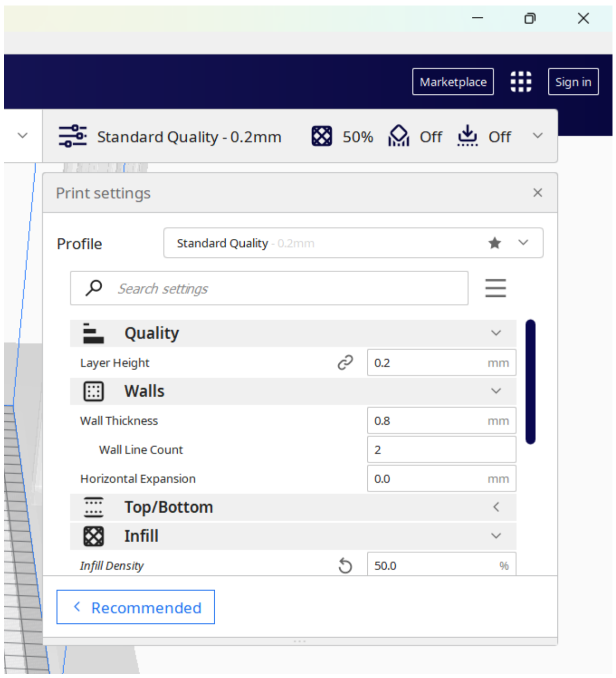


Fig 4.3 Cura settings for PLA

Generating GCODE:

Once all the settings are configured, generate the GCODE file in Cura Ultimaker.

GCODE contains the specific instructions that the 3D printer will follow to print the model.

The GCODE file includes information about layer heights, movement paths, extrusion rates, and other print parameters.

Transferring GCODE to the Printer:

Save the generated GCODE file to an SD card or transfer it to the Creality 10S Pro 3D printer through a USB connection.

Ensure that the GCODE file is compatible with the printer's firmware.

4.3 3D-Printing:

Insert the SD card with the GCODE file into the 3D printer or initiate printing through the connected computer.

Follow the printer's instructions to start the printing process.

Monitor the print progress and make any necessary adjustments or troubleshooting if required.

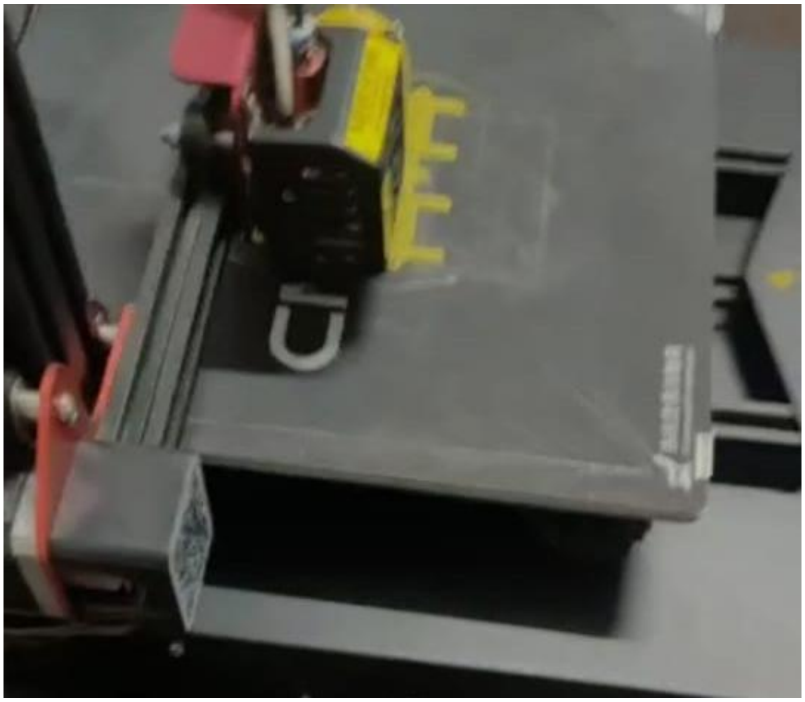


Fig 4.4 3D printing in Creality 10S pro

By following this process, you can design a quadrupedal body using FreeCAD, generate GCODE in Cura Ultimaker, and successfully print the model on the Creality 10S Pro 3D printer. It allows you to leverage the capabilities of both software tools to achieve accurate and high-quality 3D prints for your quadrupedal robot project.

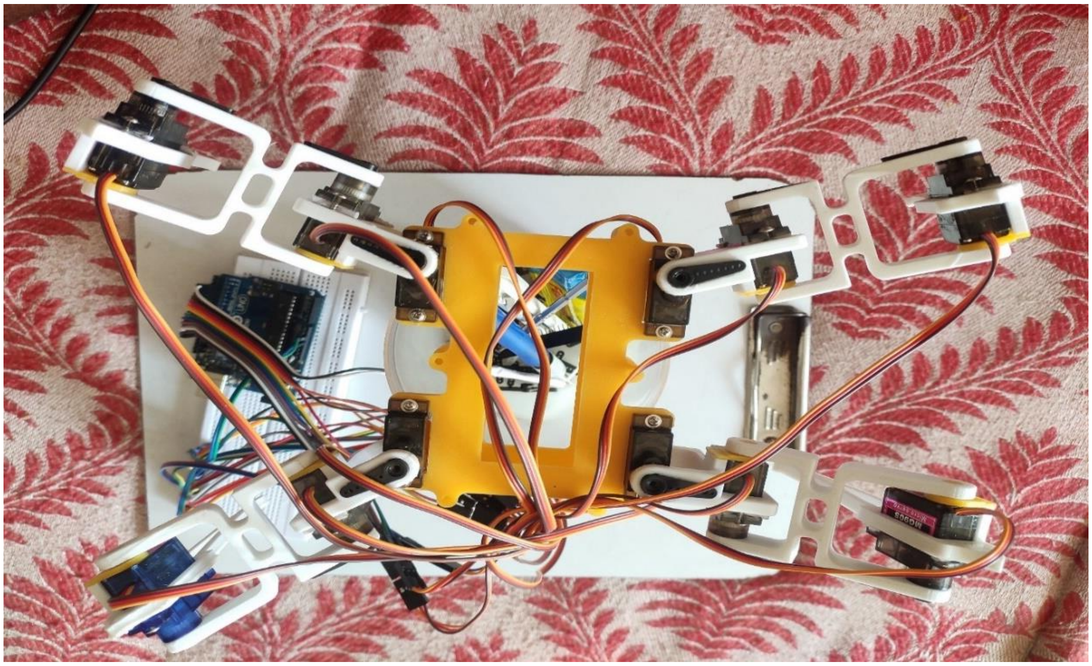


Fig 4.5 Parts after assembled

CHAPTER- V

SOFTWARE SPECIFICATION AND LANGUAGES USED

5.1 Arduino IDE:

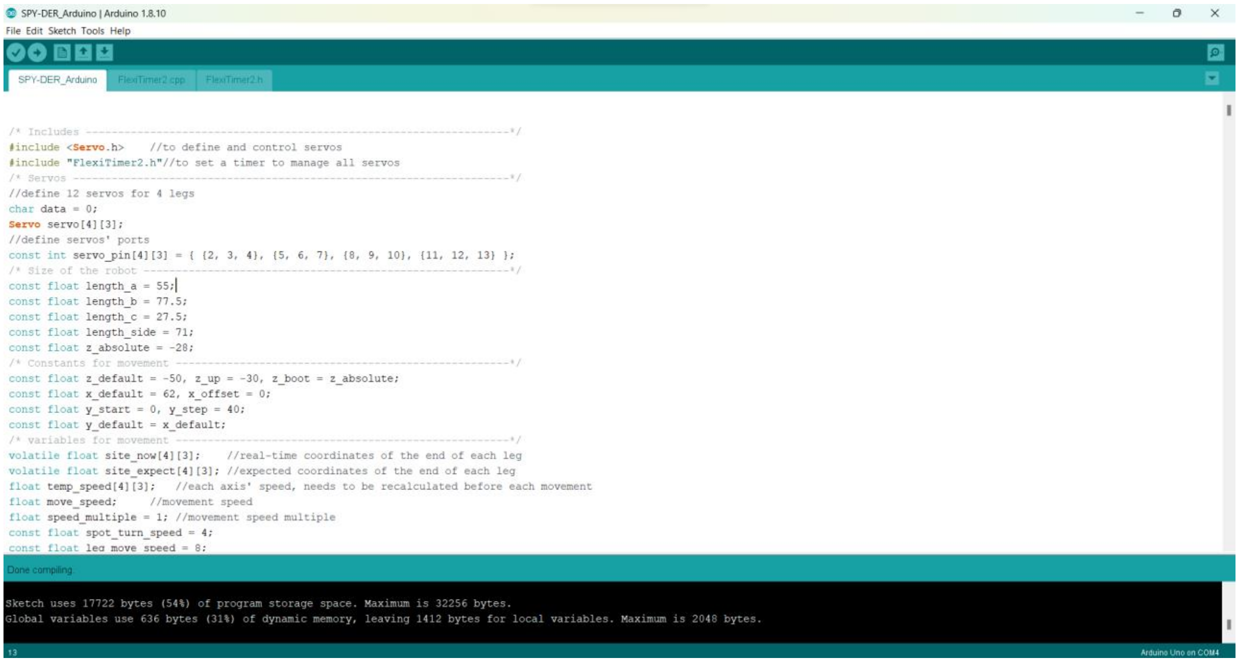


Fig 5.1 Arduino IDE

The Arduino IDE (Integrated Development Environment) is a software application used for writing, compiling, and uploading code to Arduino microcontroller boards. It provides a user-friendly interface for programming Arduino projects and supports the Arduino programming language, which is based on C/C++. The IDE offers a text editor, a compiler, and a bootloader to facilitate the development process. It includes a vast library of pre-written code examples and functions that simplify the programming of various Arduino components and sensors. The Arduino IDE is cross-platform, compatible with Windows, macOS, and Linux, making it accessible to a wide range of users.

5.2 Raspbian OS:

Raspbian is the official operating system (OS) for the Raspberry Pi, a popular single-board computer. It is a Linux-based OS specifically designed for the Raspberry Pi's ARM architecture. Raspbian is based on the Debian Linux distribution and provides a lightweight and optimized environment for running applications on the Raspberry Pi. It offers a user-friendly desktop interface, a command-line interface, and a vast software repository with a wide range of pre-installed tools and applications. Raspbian is known for its stability, flexibility, and extensive community support, making it a preferred choice for Raspberry Pi users and developers.

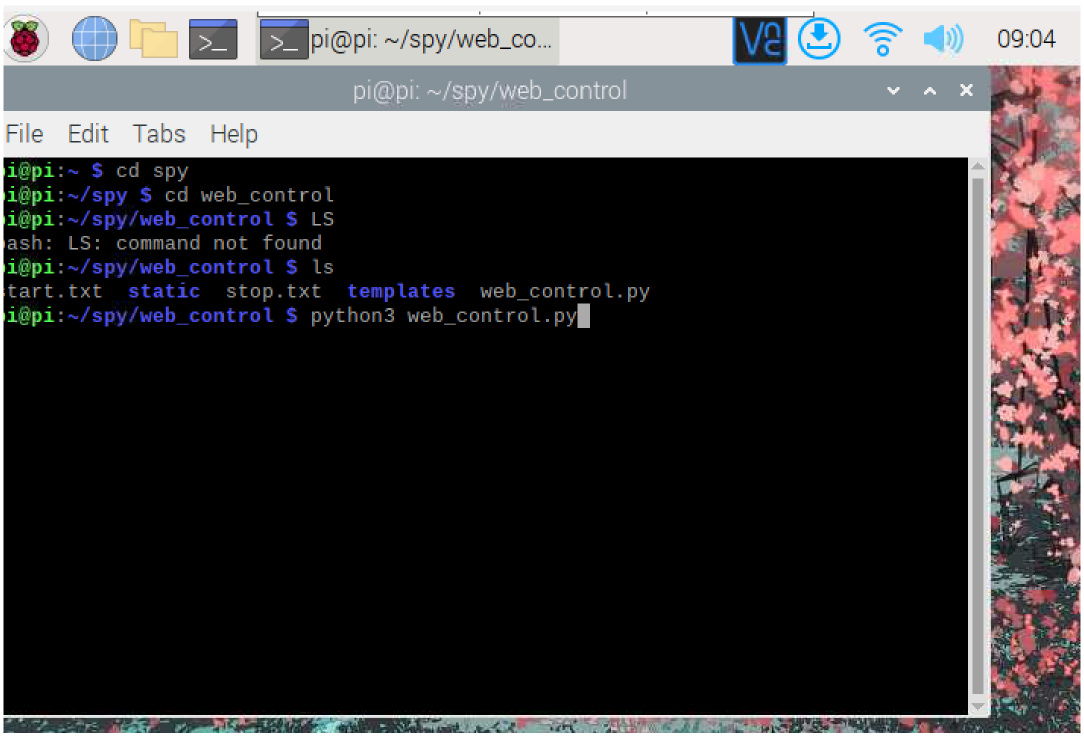


Fig5.2 Raspberry Terminal

Terminal:

The Terminal, also known as the command-line interface (CLI), is a text-based interface used to interact with the operating system and execute commands. It provides a way to control the computer or device through text commands rather than a graphical user interface (GUI). In Raspbian OS, the Terminal is accessible through the desktop interface or can be accessed remotely via SSH (Secure Shell). It allows users to navigate the file system, execute programs, modify system settings, install software, and perform various administrative tasks using command-line commands. The Terminal provides a powerful and flexible way to interact with the Raspberry Pi, perform scripting, and automate tasks. It is particularly useful for advanced users, developers, and system administrators who prefer the efficiency and flexibility of the command-line interface.

5.3 VNC Viewer:

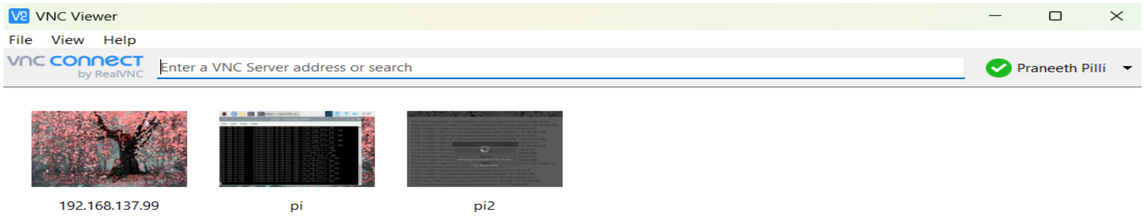


Fig 5.3 VNC Viewer

VNC Viewer is a software application that allows users to remotely access and display the graphical interface of a Raspberry Pi on another device. It enables users to control and view the Raspberry Pi's desktop environment, applications, and files from a different computer or mobile device over a network connection. By using VNC (Virtual Network Computing) technology, the viewer application establishes a connection with the Raspberry Pi, transmitting screen updates and user input in real-time. VNC Viewer provides a convenient way to monitor and interact with a Raspberry Pi without the need for a physical display, keyboard, or mouse directly connected to the Pi, making it ideal for remote administration, troubleshooting, and headless operations.

5.4 libraries used:

servo.h Library:

The servo.h library is a standard library in the Arduino IDE that provides functions for controlling servo motors. Servo motors are commonly used in robotics and other projects where precise control of angular position is required. The servo.h library simplifies the process of controlling servos by abstracting the low-level details of pulse width modulation (PWM) signals used to control the motor's position.

The servo.h library includes functions such as attach(), detach(), write(), and writeMicroseconds() that allow you to easily control the movement and position of servo motors. You can specify the desired angle or pulse width to position the servo motor accurately. The library takes care of generating the appropriate PWM signals required to control the servo.

FlexiTimer2.h Library:

The FlexiTimer2.h library is a third-party library for Arduino that provides more flexible and advanced timer functionalities compared to the built-in timer functions. It offers precise control over timer interrupts and timing intervals, allowing for custom timing configurations and scheduling of tasks.

The FlexiTimer2.h library is particularly useful in scenarios where precise timing is crucial, such as generating accurate pulse widths, driving stepper motors, or implementing time-sensitive tasks. It provides functions like setPeriod(), attachInterrupt(), start(), and stop() that enable you to configure and control the behavior of timer interrupts.

Using the FlexiTimer2.h library, you can define custom timing intervals, specify the duration of interrupts, and execute specific code at designated intervals. This library extends the timer capabilities of the Arduino, providing more flexibility and control over time-dependent operations.

Both the servo.h and FlexiTimer2.h libraries enhance the functionality and ease of use of Arduino by providing higher-level abstractions and convenient functions to control servo motors and timers respectively, enabling developers to focus on higher-level logic and functionality in their projects.

Flask:

Flask is a web framework for Python that allows the creation of web applications. It provides routing, templating, and request handling capabilities, making it easy to build web interfaces for interacting with the Raspberry Pi and Arduino. RPi.GPIO: RPi.GPIO is a library that enables control of the Raspberry Pi's GPIO (General Purpose Input/Output) pins. It provides functions to configure and control the GPIO pins, allowing interaction with external devices and sensors.

time:

The time library provides functions to work with time-related operations. In this code, it may be used for introducing delays or timing-related tasks.

serial:

The serial library facilitates communication between the Raspberry Pi and Arduino via the serial port. It provides methods for establishing a serial connection, sending/receiving data, and configuring the serial settings.

OS:

The os library provides various functions for interacting with the operating system. In this code, it is used to execute system commands for controlling the Raspberry Pi, such as starting or stopping scripts and performing a power-off action.

These libraries enhance the functionality of the application by enabling web development, GPIO control, serial communication, timing operations, and interaction with the underlying operating system on the Raspberry Pi.

CHAPTER- VI

METHODOLOGY OF THE SYSTEM

6.1 Connections of the Proposed System

The 12 MG90S servos are connected to the digital pins of Arduino Uno from pin 2 to pin13 and initialized in the following way as shown in the figure below.

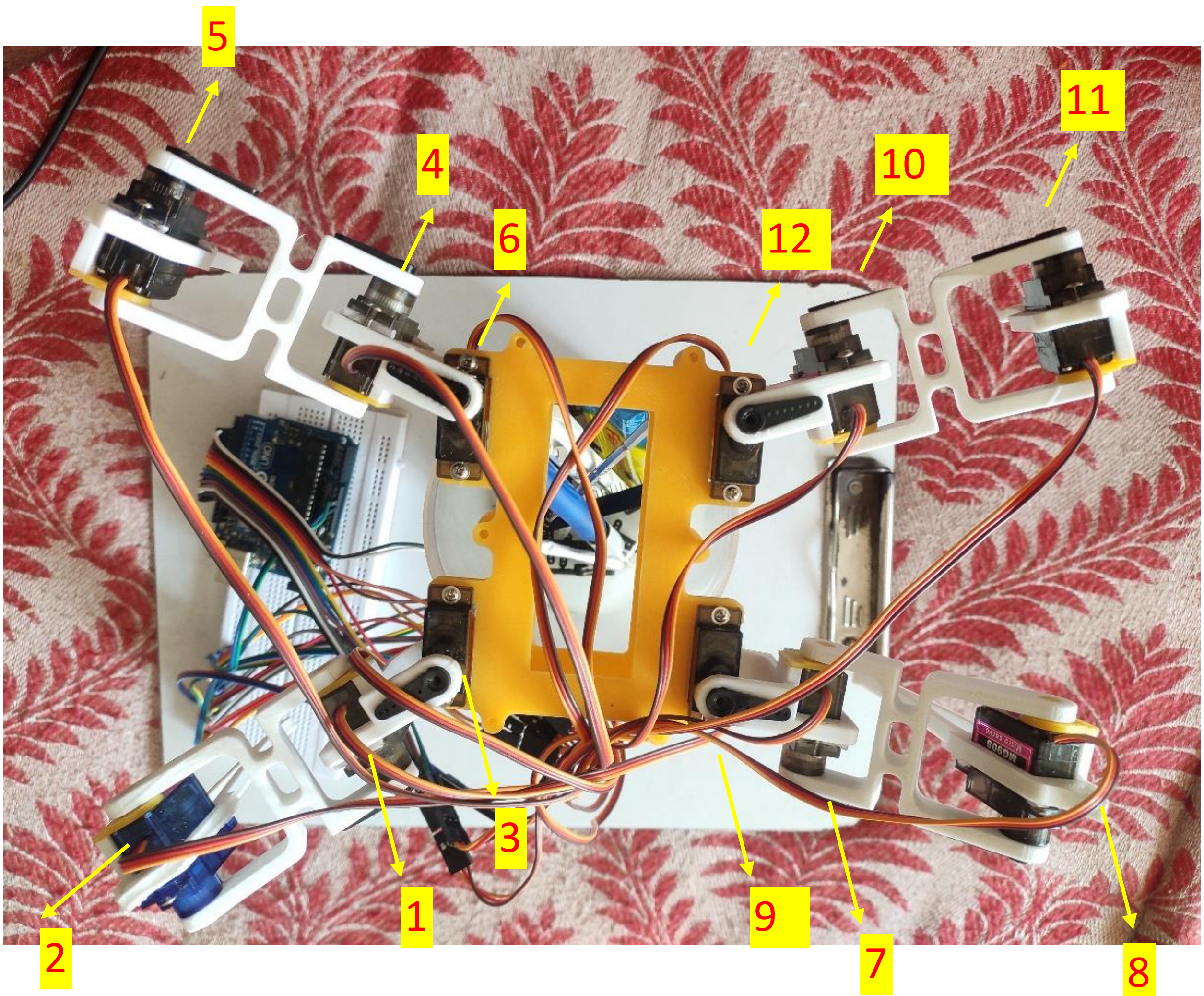


Fig 6.1 Numbering of servos

6.2 Working of Legs of bot

Each movement like go forward, backward, turn left and turn right are given a void loop and function and when the function is called via serial pin by Raspberry Pi those are TX and RX pins on Arduino Uno

6.3 Serial communication between Arduino Uno And Raspberry Pi

When connecting an Arduino Uno and a Raspberry Pi 3 via the serial pins (0 and 1), a logic level shifter is often used to ensure compatibility between the two devices. Here's an explanation of the connection setup:

Arduino Uno:

The Arduino Uno has two hardware UART (Universal Asynchronous Receiver/Transmitter) serial ports: Serial 0and Serial1. The default pins for the Serial port on the Arduino Uno are pins 0 (RX) and 1 (TX). The Serial port is used for communication with the Raspberry Pi.

Raspberry Pi 3:

The Raspberry Pi 3 also has hardware UART pins, which are located on the GPIO header. The default pins for the hardware UART on the Raspberry Pi 3 are GPIO 8 (TXD) and GPIO 10 (RXD). These GPIO pins correspond to the transmit (TX) and receive (RX) lines, respectively.

Logic Level Shifter:

Since the voltage levels of the Arduino Uno (5V) and Raspberry Pi 3 (3.3V) differ, a logic level shifter is required to ensure proper signal communication between them.

A logic level shifter can convert the 5V signals from the Arduino Uno to 3.3V signals compatible with the Raspberry Pi 3.

Connection Steps:

Connect the Arduino Uno's TX pin (pin 1) to the input side of the logic level shifter. Connect the Raspberry Pi 3's RXD GPIO pin (GPIO 10) to the output side of the logic level shifter.

Connect the Arduino Uno's RX pin (pin 0) to the input side of another logic level shifter.

Connect the Raspberry Pi 3's TXD GPIO pin (GPIO 8) to the output side of the second logic level shifter.

Ensure that the appropriate power and ground connections are made between the Arduino Uno, Raspberry Pi 3, and the logic level shifter.

By using a logic level shifter, you can safely connect the Arduino Uno and Raspberry Pi 3, allowing them to communicate via the serial pins while ensuring the voltage compatibility between the two devices.

6.4 RPi-Cam-Web-Interface:

RPi-Cam-Web-Interface is a software package designed for Raspberry Pi that allows users to control and manage a Raspberry Pi camera module remotely through a web interface. It provides a convenient way to access and control the camera module, view the live video feed, capture images, and record videos. Here are some key features and functionalities of RPi-Cam-Web-Interface:

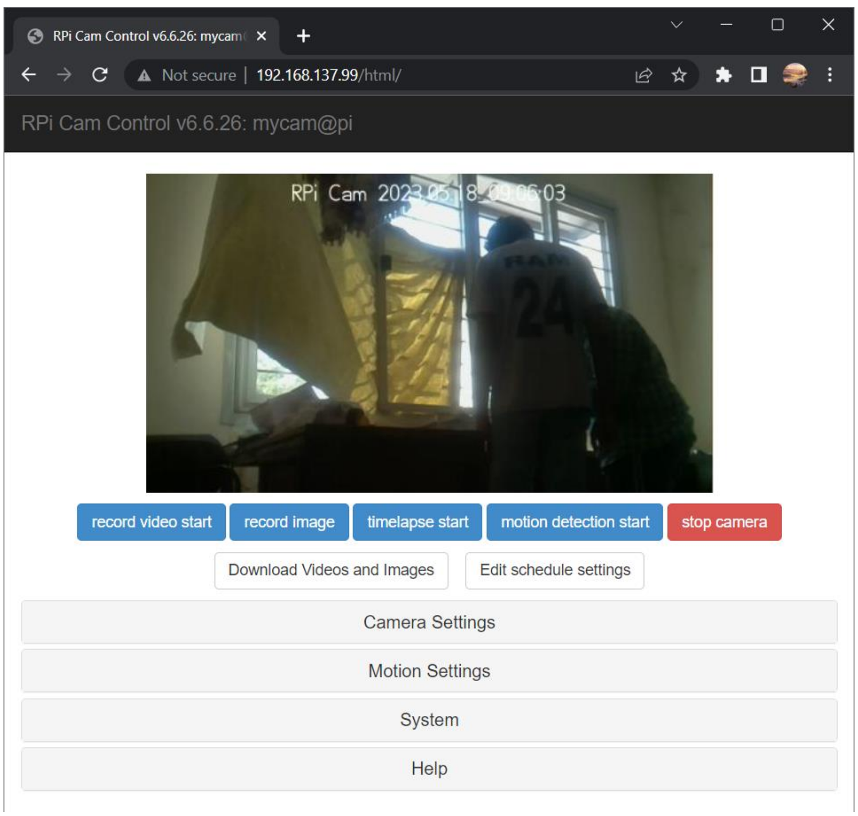


Fig 6.2 Camera Interface

Web Interface: RPi-Cam-Web-Interface sets up a web server on the Raspberry Pi, allowing users to access the camera module's functionality through a web browser on any device connected to the same network.

Live Video Streaming: The software enables real-time video streaming from the Raspberry Pi camera module to the web interface. Users can view the live video feed and monitor the camera remotely.

Capture Images: RPi-Cam-Web-Interface provides options to capture images directly from the camera module. Users can take snapshots of the live video feed or configure interval-based image capture.

Video Recording: The software supports recording videos using the Raspberry Pi camera module. Users can start and stop video recording, adjust video settings such as resolution and frame rate, and save the recorded videos.

Configuration Options: RPi-Cam-Web-Interface offers various configuration options to customize the camera settings. Users can adjust parameters such as exposure, white balance, image effects, and more through the web interface.

Motion Detection: The software includes motion detection functionality, allowing users to configure the camera module to detect motion and trigger specific actions, such as capturing images or recording videos when movement is detected.

RPi-Cam-Web-Interface simplifies the process of remotely controlling and managing a Raspberry Pi camera module, providing a user-friendly web interface for easy access and configuration. It is a popular choice for Raspberry Pi-based surveillance systems, home monitoring, and various other applications that require camera functionality with remote access.

6.5 Code execution:

The provided code appears to be a Flask application that interacts with an Arduino board connected to a Raspberry Pi. Here's a breakdown of the code:

The Flask library is imported to create a Flask application. render_template and request are imported to handle HTML templates and incoming requests. RPi.GPIO is imported to control the Raspberry Pi's GPIO pins. time, serial, and os libraries are imported for various functionalities An instance of the Flask application is created. A serial connection is established with the Arduino board using the serial.Serial method. The / route is defined to render the index.html template.

Routes are defined for different actions like left, right, forward, backward movement, stop, light on/off, hand wave, dance, camera on/off, standing, sitting, and power off.

Each route sends a corresponding command to the Arduino through the serial connection.

The application is set to run on host ' http://192.168.137.99/' which is address of the raspberry pi that's being connected to the specific network and port 5010 when executed directly.

It's important to note that the code assumes the Arduino board is connected to the Raspberry Pi through the /dev/ttyAMA0 serial port and expects specific commands to be sent to control the connected hardware. Additionally, it interacts with the Raspberry Pi's camera using the /home/pi/RPi_Cam_Web_Interface/ script. Ensure that the necessary dependencies, configurations, and hardware connections are properly set up before running this code.

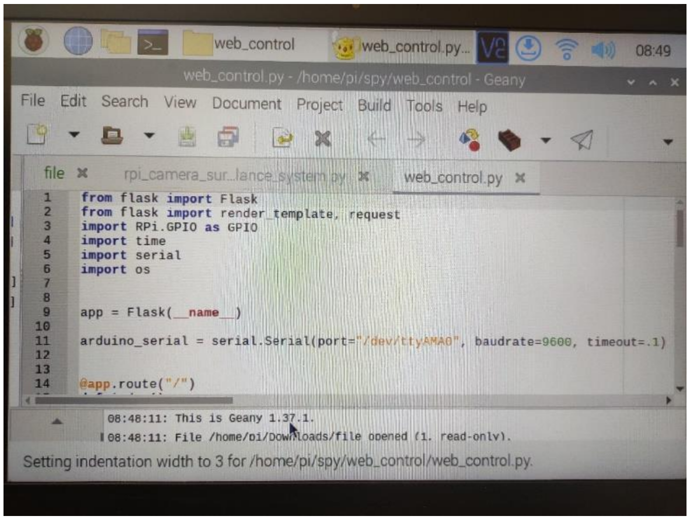


Fig 6.3 libraries used

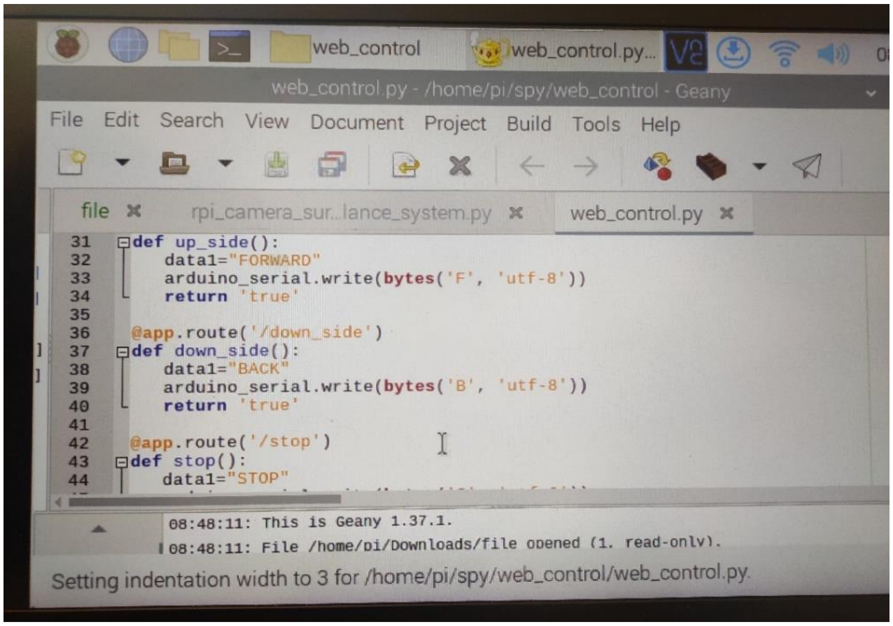


Fig 6.4 initializing the variable to each movement

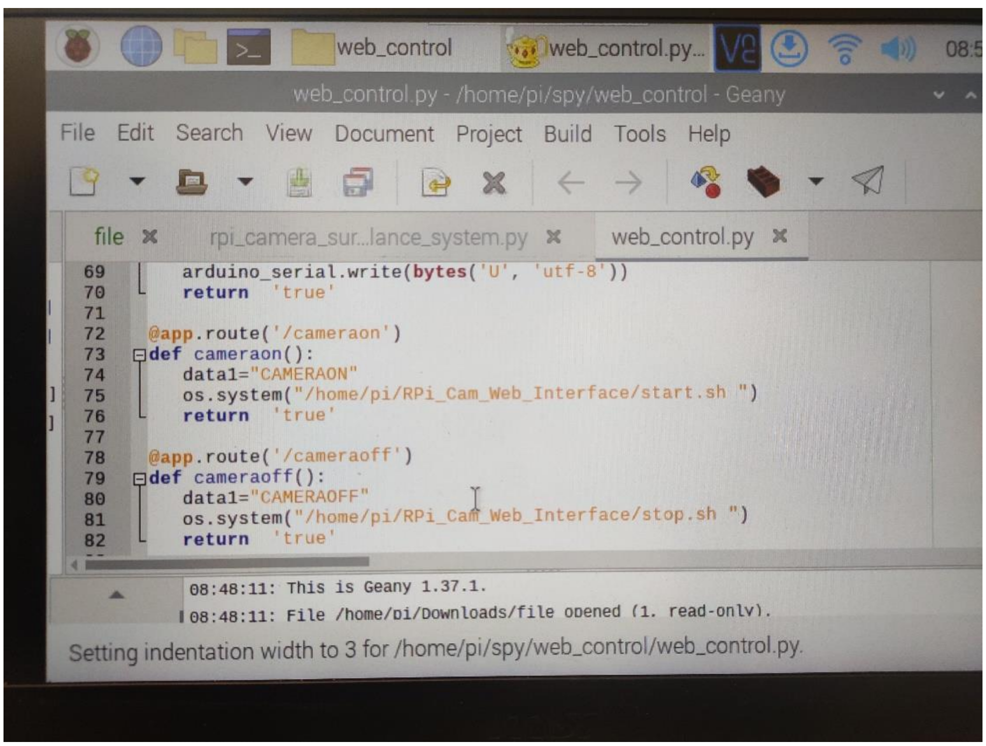


Fig 6.5 path for pi camera control module

CHAPTER- VII

RESULTS

7.1 Final Prototype:

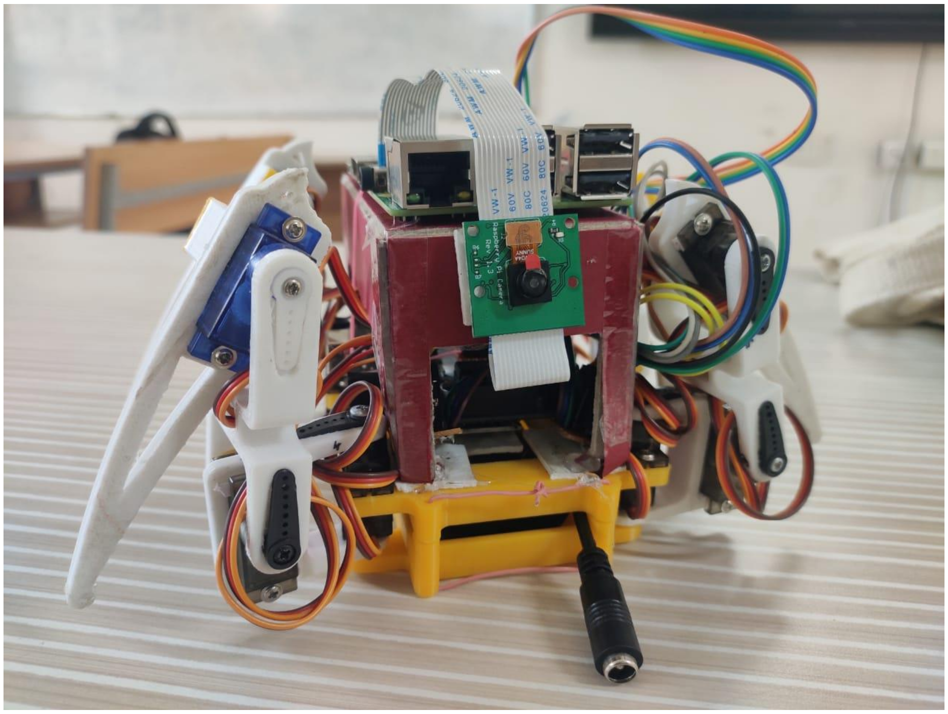
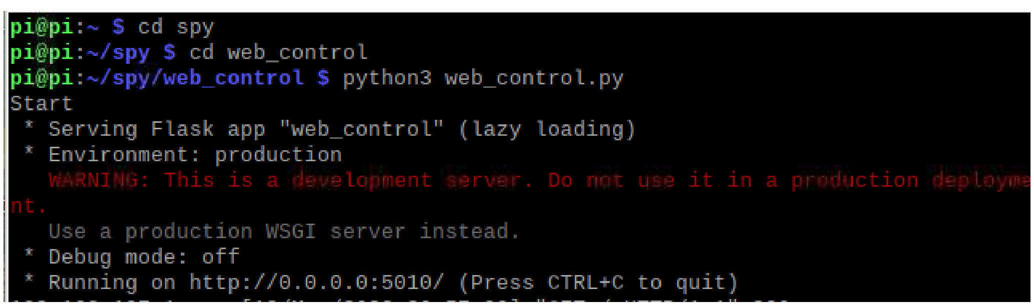


Fig 7.1 Final prototype of Quad-bot

After Connecting all the servos and Arduino with the power source and raspberry pi .The camera is positioned at right angle to capture properly.

7.2 Execution of Raspberry system:



```
pi@pi:~ $ cd spy
pi@pi:~/spy $ cd web_control
pi@pi:~/spy/web_control $ python3 web_control.py
Start
 * Serving Flask app "web_control" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:5010/ (Press CTRL+C to quit)
```

Fig7.2 commands to execute the code in pi folders

The raspberry starts a web page where the controls of the bot are initialized in the form of button as shown below.

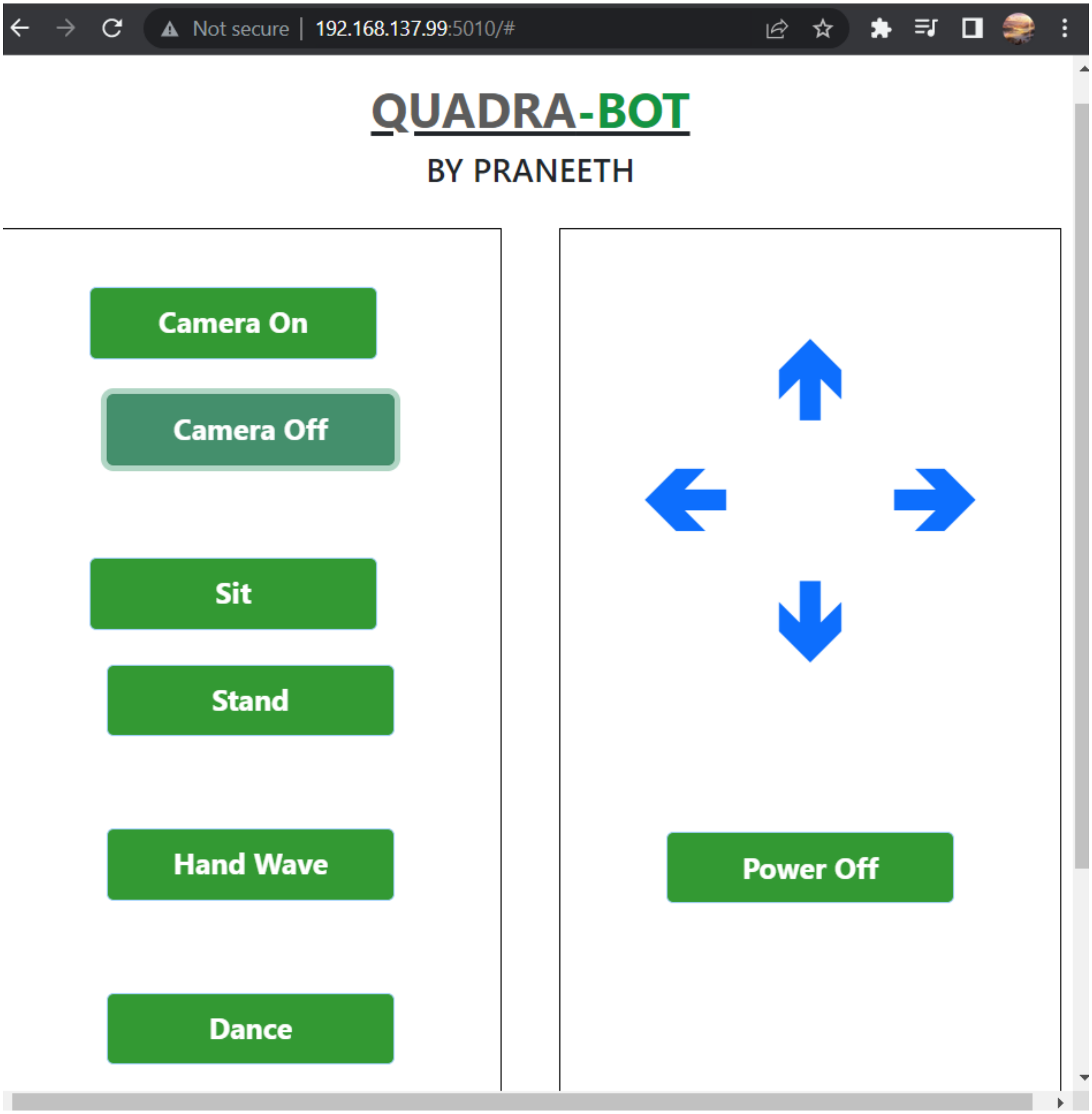


Fig7.3 Web page for the controls

7.3 Combined function of web control and surveillance camera:

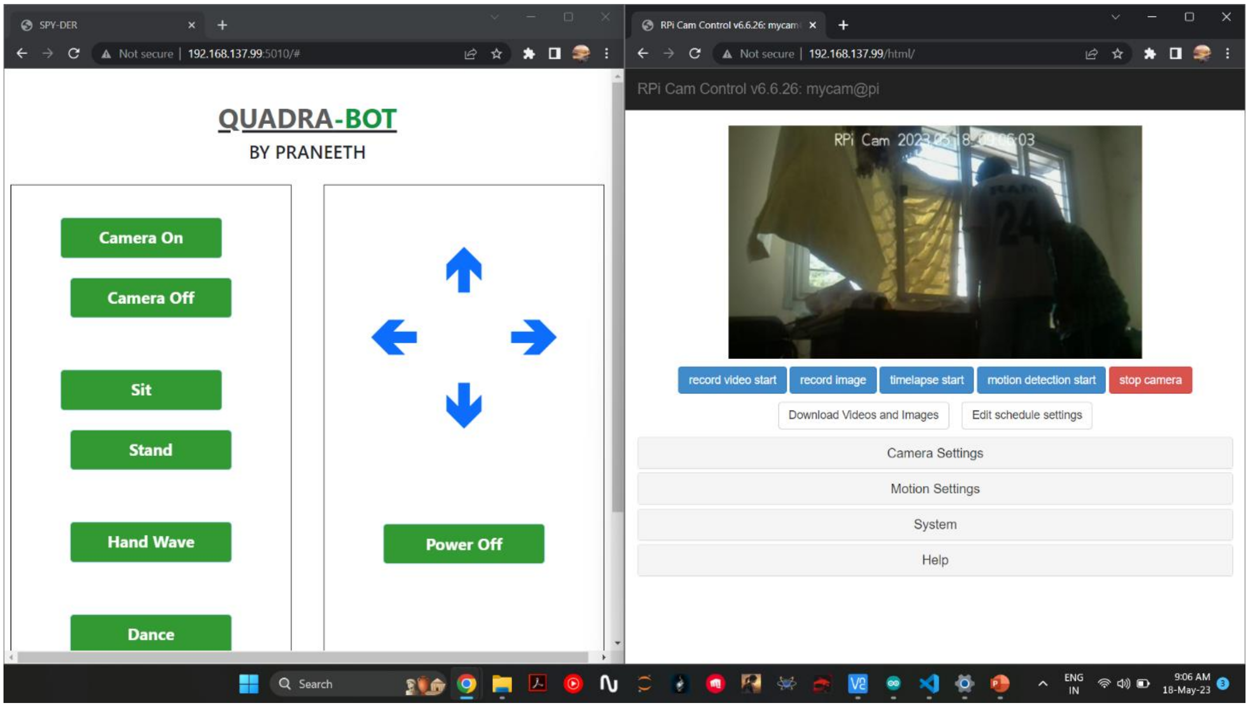


Fig 7.4 final control of Quad-bot

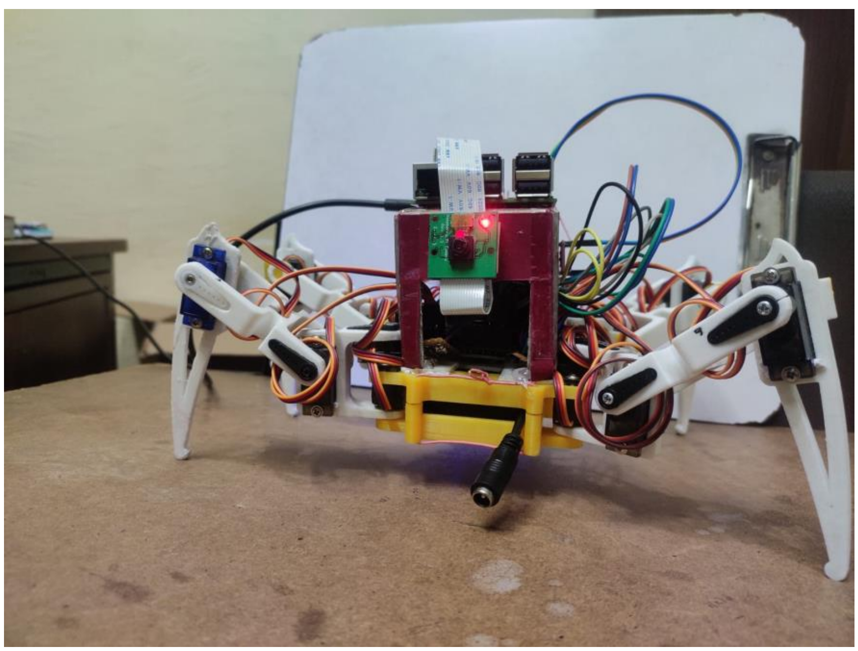


Fig7.5 bot moving forward with camera on

CHAPTER- VIII

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion:

The development of the Quadrupedal surveillance robot using a Raspberry Pi and Arduino Uno proved to be a successful endeavor. The project highlighted the seamless integration of hardware and software components, showcasing the capabilities of these platforms in the realm of robotics.

The Raspberry Pi served as the brain of the robot, enabling high-level control and hosting a web interface for remote operation. Through the web interface, users could effortlessly command the robot's movements and access its various features. This showcased the power and versatility of the Raspberry Pi as a control unit for robotics applications.

The Arduino Uno played a crucial role in controlling the servo motors responsible for leg movement. The Servo library provided convenient functions for servo control, while the FlexiTimer2 library ensured precise timing of servo commands. The integration of these libraries with the Arduino Uno facilitated smooth and accurate leg movements, contributing to the overall performance of the robot.

The utilization of 3D designing software like FreeCAD and slicing software like Cura Ultimaker allowed for the creation of the robot's body using a 3D printer. This demonstrated the potential of additive manufacturing techniques in building customized and intricate robot components.

The successful implementation of the project opened up exciting possibilities for Quadrupedal robots in surveillance and monitoring scenarios. The robot's ability to move in different directions, turn, control a camera, and perform predefined behaviors showcased its potential applications in security, exploration, and remote monitoring.

Furthermore, the project highlighted the importance of hardware integration and software development skills in the field of robotics. The seamless communication between the Raspberry Pi and Arduino Uno, facilitated by the logic level shifter, exemplified the significance of proper connectivity and coordination among different hardware components.

In conclusion, the Quadrupedal surveillance robot project demonstrated the capabilities of the Raspberry Pi and Arduino Uno in creating a functional and remotely controlled robot. The integration of hardware, software, and 3D printing techniques showcased the potential for further advancements in robotics. The project's success not only provided a valuable learning experience but also laid the foundation for future explorations and innovations in the field of robotics and automation.

8.2 Future scope:

This project on the Quadrupedal surveillance robot using a Raspberry Pi and Arduino Uno lays the foundation for several future scope and advancements. Here are some potential areas for further development and improvement:

1. **Autonomous Navigation:** Implementing algorithms and sensors such as ultrasonic sensors, cameras, or LIDAR can enable the robot to navigate autonomously, avoiding obstacles and mapping its surroundings. This would enhance its surveillance capabilities and make it more versatile in various environments.
2. **Wireless Communication:** Integrating wireless communication modules such as Bluetooth or Wi-Fi can enhance the robot's remote-control capabilities. It would allow for wireless interaction and real-time monitoring of the robot, enabling more flexible and convenient operation.
3. **Machine Learning and AI:** Incorporating machine learning and AI techniques can enable the robot to learn and adapt to its environment. This could involve developing algorithms for object recognition, gesture detection, or even behavior learning, enhancing the robot's ability to perform tasks and interact intelligently with its surroundings.
4. **Enhanced Sensor Integration:** Integrating additional sensors like temperature sensors, humidity sensors, or gas sensors can expand the robot's surveillance capabilities. These sensors can provide valuable environmental data, enabling the robot to detect anomalies or monitor specific conditions.
5. **Extended Battery Life:** Optimizing power management systems and utilizing energy-efficient components can extend the robot's battery life. This would allow for longer operational periods, making the robot more suitable for prolonged surveillance missions or remote monitoring applications.
6. **Improved Mechanical Design:** Refining the mechanical design of the robot's body can enhance its stability, agility, and overall performance. This could involve exploring

lightweight materials, advanced joint mechanisms, or implementing active stabilization techniques for smoother movements.

7. **Cloud Integration:** Integrating the robot with cloud platforms can enable data storage, analysis, and remote access to captured images and videos. It would facilitate seamless data sharing, remote monitoring, and enable advanced analytics for surveillance applications.
8. **Swarming and Cooperative Behavior:** Developing algorithms for multiple robots to communicate and collaborate can enable swarming behavior and coordinated surveillance missions. This would allow a group of robots to cover larger areas efficiently and cooperate in performing complex tasks.
9. **User Interface Enhancements:** Improving the web interface with advanced visualization features, intuitive controls, and additional functionalities can enhance the user experience and make robot control more user-friendly.
10. **Customization and Modularity:** Designing the robot in a modular and customizable manner would allow for easy upgrades, component replacements, and the integration of additional features based on specific requirements.

These future scope and advancements can enhance the capabilities and functionality of the Quadrupedal surveillance robot, making it more efficient, intelligent, and versatile for various surveillance and monitoring applications.

8.3 Advantages:

1. **Versatility:** The robot's quadrupedal design enables it to navigate a wide range of terrains, making it suitable for various surveillance and monitoring applications. It can traverse uneven surfaces, climb stairs, and maneuver through obstacles, allowing it to access different locations with ease.
2. **Remote Operation:** The integration of a web interface hosted on the Raspberry Pi enables remote control and monitoring of the robot. Users can access the interface from any device with a browser and command the robot's movements, camera, and other features from a distance, providing flexibility and convenience.
3. **Surveillance Capabilities:** The robot's ability to carry a camera and perform movements controlled by the user enables it to serve as an effective surveillance tool. It can be deployed

in areas where human access is limited or risky, allowing for remote monitoring of spaces, objects, or events in real-time.

4. **Customizability:** The use of 3D designing software and a 3D printer allows for customization of the robot's body and components. This enables adaptation to specific requirements, such as integrating additional sensors, payload capacity, or aesthetic modifications.
5. **Educational Value:** The project provides a hands-on learning experience in robotics, electronics, and programming. It allows enthusiasts, students, and hobbyists to gain practical knowledge in building and controlling a robotic system, fostering skills in hardware integration, software development, and problem-solving.
6. **Integration of Raspberry Pi and Arduino:** The combination of Raspberry Pi and Arduino Uno brings together the computational power and versatility of the Raspberry Pi and the real-time control capabilities of the Arduino. This allows for a seamless integration of high-level control, web hosting, and sensor interfacing, creating a powerful and flexible robotic platform.
7. **Scalability:** The project lays the foundation for scalability, allowing for future expansion and advancements. Additional features, sensors, or behaviors can be incorporated into the robot's design, enabling further customization and capabilities based on specific requirements.
8. **Low-Cost Solution:** The project utilizes affordable and easily accessible components like the Raspberry Pi, Arduino Uno, and servo motors, making it a cost-effective solution compared to commercial surveillance robots. This makes it accessible to a wider range of users with limited budgets.
9. **Open-Source Community:** The project leverages the open-source nature of the Raspberry Pi, Arduino, and associated libraries, benefiting from a vast online community. Users can access extensive documentation, tutorials, and forums, allowing for knowledge sharing, troubleshooting, and further development.
10. **Potential for Innovation:** The project encourages exploration and innovation in the field of robotics. It serves as a starting point for enthusiasts to experiment, refine, and expand the capabilities of the Quadrupedal surveillance robot, leading to potential advancements, new applications, and breakthroughs in the domain.

Overall, the project offers advantages in terms of versatility, remote operation, surveillance capabilities, customizability, educational value, integration of platforms, scalability, cost-effectiveness, access to the open-source community, and potential for innovation.

8.4 Limitations and challenges:

While the Quadrupedal surveillance robot project using a Raspberry Pi and Arduino Uno has various advantages, it also comes with certain limitations and challenges:

1. **Power and Battery Life:** The robot's power source and battery life can be a limitation. Depending on the size and weight of the robot and the power requirements of its components, it may have limited operating time before needing to recharge or replace the batteries. Finding a balance between power consumption and desired functionality is crucial.
2. **Payload Capacity:** The robot's payload capacity may be limited, which restricts the size and weight of additional sensors or equipment that can be integrated. This limitation can affect the range of surveillance or monitoring capabilities that can be achieved.
3. **Terrain Adaptation:** While the quadrupedal design allows for better mobility on uneven terrain compared to wheeled robots, there may still be limitations in certain challenging environments. Steep slopes, slippery surfaces, or very rough terrains can pose difficulties for the robot's stability and navigation.
4. **Communication Range:** The range of communication between the Raspberry Pi and the remote-control device can be limited by the wireless technology being used. If the robot operates outside the range of the wireless network, the control and monitoring capabilities may be affected.
5. **Real-time Responsiveness:** The latency in the communication between the Raspberry Pi, Arduino, and the servo motors can result in a delay in real-time responsiveness. This can impact the precision and agility of the robot's movements, particularly in dynamic or time-critical scenarios.

These limitations and challenges highlight the importance of thorough planning, consideration of project requirements, continuous testing and improvement, and addressing safety concerns to enhance the effectiveness and reliability of the Quadrupedal surveillance robot.

References

[1] Kalouche, A., & Bidaud, P. (2019). Design and control of quadruped robots: A review. Robotics and Autonomous Systems, 118, 43-57.

[2] Li, C., Wang, Z., Wang, X., Zhang, Y., & Xu, H. (2020). Dynamic modeling and adaptive walking control for quadruped robots: A review. Journal of Bionic Engineering, 17(5), 871-894.

[3] Cully, A., Clune, J., Tarapore, D., & Mouret, J. B. (2015). Robots that can adapt like animals. Nature, 521(7553), 503-507.

[4] Kim, S., Kim, S. W., Yim, S., & Kim, J. (2019). Design and control of a versatile quadruped robot for advanced locomotion tasks. IEEE Transactions on Robotics, 35(1), 157-172.

[5] Ham, R., Or, S., van der Smagt, P., & Ijspeert, A. (2016). A review of learning pattern generators for quadruped locomotion. Frontiers in Robotics and AI, 3, 13.

[6] Spröwitz, A., Tuleu, A., Vespignani, M., Ajallooeian, M., Badri, E., & Ijspeert, A. J. (2013). Towards dynamic trot gait locomotion: Design, control, and experiments with Cheetah-cub, a compliant quadruped robot. International Journal of Robotics Research, 32(8), 932-950.

[7] Yi, Z., Chen, X., & Gao, Y. (2019). A survey on quadrupedal robots: Practical approaches, challenges, and applications. International Journal of Advanced Robotic Systems, 16(6), 1729881419880752.

body_final_report

ORIGINALITY REPORT

8%	6%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Middlesex University Student Paper	1%
2	Submitted to University of Central England in Birmingham Student Paper	1%
3	en.wikipedia.org Internet Source	1%
4	Ryan Krauss. "Combining Raspberry Pi and Arduino to form a low-cost, real-time autonomous vehicle platform", 2016 American Control Conference (ACC), 2016 Publication	1%
5	Submitted to Indian Institute of Technology-Bhubaneswar Student Paper	<1%
6	e-journal.usd.ac.id Internet Source	<1%
7	www.shopping.com Internet Source	<1%
	www.rhydolabz.com	

8	Internet Source	<1%
9	automaticaddison.com Internet Source	<1%
10	helloraspberrypi.blogspot.tw Internet Source	<1%
11	repository.untar.ac.id Internet Source	<1%
12	Submitted to Ghana Technology University College Student Paper	<1%
13	shop.sb-components.co.uk Internet Source	<1%
14	Submitted to University of Glasgow Student Paper	<1%
15	Submitted to University of Nottingham Student Paper	<1%
16	docs.arduino.cc Internet Source	<1%
17	Charles Bell. "Windows 10 for the Internet of Things", Springer Science and Business Media LLC, 2016 Publication	<1%
18	Submitted to Roehampton University Student Paper	<1%

19	"Robot Intelligence Technology and Applications 5", Springer Science and Business Media LLC, 2019 Publication	<1 %
20	hdl.handle.net Internet Source	<1 %
21	Mike O. Ojo, Stefano Giordano, Gregorio Procissi, Ilias N. Seitanidis. "A Review of Low-End, Middle-End, and High-End Iot Devices", IEEE Access, 2018 Publication	<1 %

Exclude quotes On
Exclude bibliography On
Exclude matches Off

Contact/Personal Details of the Student

Name : Pilli Venkata Praneeth

Father's Name : P V V Padmanabham

Mother's Name : Lakshmi Bhavani

Nationality : Indian

Education Details : Bachelor of Technology
(Electronics and Communication Engineering)

Postal Address : 216-D Sector-8 , Ukkunagaram Steel Plant ,
Visakhapatnam , Andhra Pradesh – 530 032

Contact Email ID: prannu333@gmail.com
pvpraneeth@cuk.ac.in

Mobile Number: 9065286509

(Signature of the Student)

Pilli Venkata Praneeth

Place: Kadaganchi

Date: